Explainable multi-hop dense question answering using knowledge bases and text

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Somayeh Asadifar[a], Mohsen Kahani [a,*] and Saeedeh Shekarpour[b]
[a]*Ferdowsi University of Mashhad, Iran*
[b]*Department of Computer Science, University of Dayton,Dayton, Ohio*

Abstract. Much research has been conducted extracting a response from either text sources or a knowledge base (KB). The challenge becomes more complicated when the goal is to answer a question with the help of both text and KB. In these hybrid systems, we address the following challenges: i) excessive growth of search space, ii) extraction of the answer from both KB and text, iii) extracting the path to reach to the answer. A heterogeneous graph is utilized to tackle the first challenge guided by question decomposition. The second challenge is met with the usage of the idea behind an existing text-based method, and its customization for graph development. Based on this method for multi-hop questions, an approach is proposed for the extraction of answer explanation to address the third challenge. Evaluation reveals that the proposed method has the ability to extract answers in an acceptable time, while offering competitive accuracy and has created a trade-off between performance and accuracy in comparison with the base methods.

Keywords: Information Retrieval, Explainable Question Answering, Question Decomposition, Multi-hop Dense Retrieval, Knowledge-Based and Corpus

1. Introduction

In recent years, as information exchange has become pervasive through interfaces such as World Wide Web, large volumes of information are generated daily and made publicly available to everyone. The most important challenge that arises with this volume of information is finding the information needed.

Information retrieval (IR) methods are used as the core of many real-world applications. The goal of an IR system is to find documents containing the answer to the query. The purpose of a question answering (QA) system, on the other hand, is to provide answer (not just in the form of documents). Therefore, QA is closely related to other fields, such as natural language processing (NLP) and machine learning (ML).

In early researches, open-domain QA were performed to extract the answers to a question, expressed in a natural language, by a combination of manual rules and machine learning models. Recently, research has been shifted to the use of deep neural network approaches [1]–[4].

This domain includes two active research areas: knowledge-based (KB) and text-based.

KB-based models fall into two general categories: semantic parsing [5]–[8] and information retrieval [2], [3], [9]. Text-based systems include two basic modules for retrieving candidates and analyzing them to extract the correct answer [10]. Improving the efficiency of candidate retrieval is current active research. Some research performed this action with term-based TF-IDF and BM25 methods [11], [12].

[*]Corresponding author. E-mail: kahani@um.ac.ir.

Table. 1. A few questions based on the type of answer and the source needed to answer.

Row	Question	Answer	Answer Type	Source
1	What types are the films directed by the director of "For Love or Money"?	Action, Comedy, Western, Thriller, Crime	Entity	KB:Wikimovies Text: -
2	which genes are associated with diseases whose possible drugs target Cubilin?	MMAB, MUT, MMAA, MTHFD1	Entity	KB:Diseasome, Sider Text: -
3	Who composed the music for the film that depicts the early life of Jane Austen?	Adrian Johnston	Entity	KB: DBpedia Text: Wikipedia
4	What year did Guns N Roses perform a promo for a movie starring Arnold Schwarzenegger as a former New York Police detective?	1999	Non-entity	KB: DBpedia Text: Wikipedia

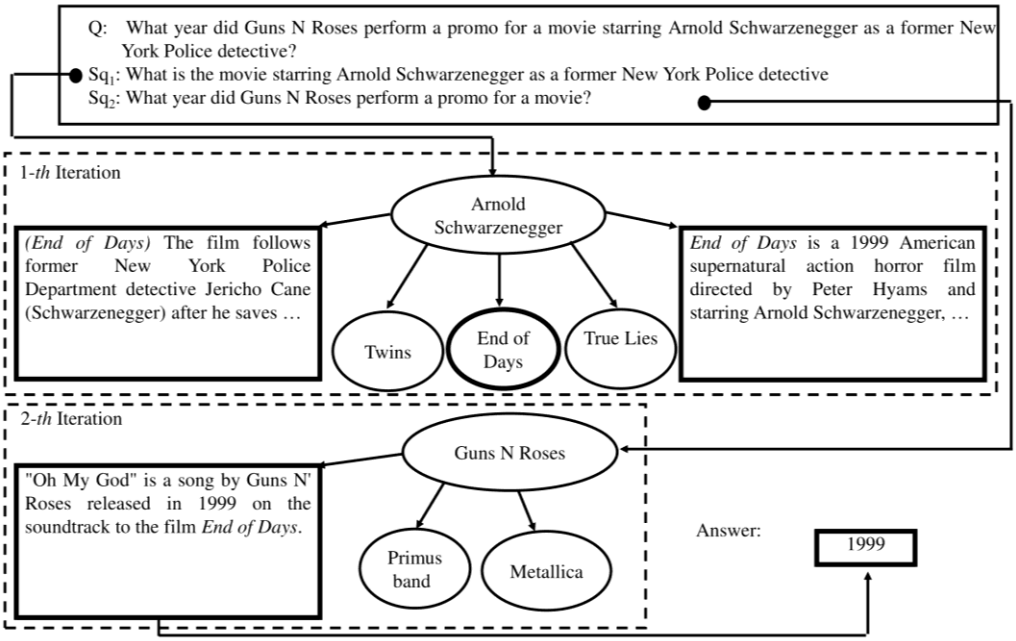

Fig. 1. An example to illustrate the process of extracting the final answer. Bold shapes indicate the entities or passages containing intermediate answers (the path to reach the final answer or explanation).

In addition, vector space methods have been proposed to solve the problems of the previous term-based approaches [13]. They are designed to increase the efficiency of the QA systems in dense retrieval methods [14], [15]. A new trend is processing questions that require multi-step inference (multi-hop question) [10], [14], [16].

Scalability and interpretability are two inseparable features in multi-hop question processing that need special attention. Addressing these issues has been successful by using dense vectors in a text-based system for multi-hop questions [15].

As mentioned, previous systems have used only one type of source (KB or text) to extract the answer (Refer to questions 1 and 2 in Table 1).

In recent years, hybrid systems have been employed to extract answers from both text and KB information sources [2], [3] (Refer to questions 3 and 4 in Table 1). Some hybrid systems can handle multi-hop questions, as well. For instance, GRAFT-Nets [2] and its expansion, PullNet [3], extract answers by building a heterogeneous graph from text, fact, and entity nodes. There are challenges in existing hybrid systems that this research seeks to address. A major challenge in existing hybrid systems is that they only have the ability to extract responses in the form of entities within the KB. For example, the answer to question 4 in Table 1 is non-entity and can only be extracted from a passage in text corpus. Existing hybrid systems are not able to answer this type of question [2], [3].

Another important challenge of existing hybrid systems is the ability to explain, which means explaining how to arrive at the final answer. Today, in addition to producing the correct answer, the explicability is important, which will lead to system transparency and trust [17], [18]. Figure 1 shows how to achieve the answer in different steps in the shapes with a thick box.

The present research is based on the PullNet approach. However, unlike PullNet, the proposed approach does not limit the extraction of the answer to the entity form in KB, only. In addition, the current method can interpret the answer with a smaller graph, while providing competitive accuracy with PullNet, by considering sub-questions and their execution order.

The proposed method also relies on MDR [15] to extract the best sequence of responses, which could provide answer explanation. Step-by-step search with the help of query decomposition and simultaneous use of KB and text leads to higher speed and accuracy compared to the MDR in extracting answers.

In the current study's scenario, one entity-linked corpus and one KB are available, neither is enough to answer the questions that require multi-hop inferences from both sources. The answer can also be extracted from KB entities and text passages, and finally, an explanation is provided on how the answer is extracted.

The main contributions of this research are:

- The ability to extract an answer from both text and KB sources, if available.
- Improving the efficiency in information retrieval as well as question answering with higher accuracy and speed by using question decomposition.
- Increasing the accuracy and speed in extracting answers by using text and KB, simultaneously.
- The ability to extract answer explanation.

The rest of the paper is organized as follows. Section 2 describes the research objectives and questions that are answered in this article. Section 3 reviews related works. The Backgrounds and proposed approach are presented in Section 4 and 5, respectively. Sections 6 provides the experimental and empirical results. In the discussion section, the findings, theoretical and practical implications of the research are reviewed. Section 8 presents the conclusion and future works. Some sample questions and answers are provided in the appendix for further explanation.

2. Research Objectives

Our main goal in this research is to provide a question answering system that can answer multi-hop questions using two sources of structured and unstructured (hybrid approach).

In this study, specifically, the research objective is to address the following research questions:

- Research question 1 (RQ1): What is the effect of question decomposition on the accuracy and speed of answer extraction in hybrid QA systems? (§ 5.2, Table 5 and Table 6)
- Research question 2 (RQ2): What is the relationship between the number of sub-questions and the number of answer search steps? (§ 7.1, first finding)
- Research question 3 (RQ3): How to get the sequence of intermediate answers (explanation), including text passages and knowledge base triples (hybrid systems)? (§ 5.1.3)

3. Related Works

This open domain QA field has a long history of being characterized by two parts: KB-based and text-based. A new field has emerged by combining these two models, hybrid. In this section, the previous works in these areas are explored in some details. The ability to extract explanation to answer hybrid questions is one of the main axes examined in this article. Therefore, in this section, works related to this field are also discussed.

3.1. KB-based

The literature in this category is based on two general methods: semantic parsing and Information Retrieval-Based.

Semantic parsing methods: These methods are based on predefined patterns or rules for converting input questions into the logical forms. The limited number of patterns has restricted the ability to answer complex questions in these systems. To remedy this problem, Abujabal et al. have developed a semi-automatic pattern learning approach [5].

Recently, systems have tended to use neural networks to increase efficiency and scalability. The question in a natural language is first transformed into an intermediate representation, such as a tree or graph. Then, the intermediated representation is converted to a logical form. Efforts have been made to increase the efficiency in identifying entities [19], generating question graph [20], and processing multi-hop questions [6], [8].

Efforts have been made to increase the efficiency of the components of identifying entities [19] and scoring

the generated graphs [20], as well as to solve the problem of answering multi-hop questions [6], [8]. Performing semantic analysis using encoder-decoder models is another method that has become common in recent years. These methods, which often use a Long Short-Term Memory (LSTM) network, differ in choosing a decoder, whether a tree (Seq-to-Tree) or a sequence (Seq-to-Seq) [21]. They generally ignore the structural information and interdependence between question words by capturing only the order of the words. This problem was addressed by combining the tree structure and sequence in a graph representation using Graph-to-Seq model [7].

The methods described above, although practical, require a large amount of training data, which is costly to generate. In order to solve this challenge, efforts were made to develop methods with weaker supervision, for example, using reinforcement learning [22].

Information Retrieval-Based Methods: In these methods, first, the desired entities (i.e., topic entity) in the question are captured, then the entities are linked to the knowledge base. In the next step, subgraphs containing the desired entities are selected from the KB. Nodes in the subgraphs other than question entities are considered as answers.

In early researches [23], the classification of syntactic features of the candidate's questions and answers were employed. This not only was time-consuming but also did not include all semantic features, as these features were defined manually. To solve the problem, research in this field turned into representation learning of question. Questions and candidate answers were represented in the vector space, and later, neural networks are used for better representation.

In addition, topic entity, the path between topic and answer entities, context (KB subgraph containing the topic entity), and answer entity are often employed to represent the candidate response [24].

For instance, the cross-attention-based neural network model captures the correlation between questions and answers and uses the above four mentioned properties to encode candidate answers. It has reported acceptable performance [9] compared to the previous approaches.

A method based on learning graph representation is presented in [2], [3]. These two works utilize the text body, in addition to the KB, to find the answer, and are classified in the hybrid category. In these methods, heterogeneous graphs are extracted from knowledge base entities and body texts. Learning is performed using a convolutional Neural Network (CNN), and a classifier determines the answer. The present study is inspired by the graph representation learning method in these articles [2], [3].

The trend of knowledge-based information retrieval approaches has recently tended to process multi-hop questions that infer answers in two ways: using memory networks and walking the path. For the former, one can refer to the work proposed by Chen et al. [25], which uses a bidirectional attentive memory network that utilizes the correlation factor to improve the representation of the question.

For the second category, Qiu et al. look at the QA issue as a matter of sequential decisions [26]. In this research, a stepwise reasoning network has been created, which is trained using reinforcement learning.

Although these methods do not use predetermined patterns, they still have difficulty processing complex questions, and most methods are challenging in creating interpretation.

Recently, KB-based systems have provided solutions for processing complex questions that include multiple entities, relationships, and constraints, often in the form of a sequence of questions. These questions can be answered by breaking into simple questions, based on question syntax and predefined templates [5], [6], [27] or question semantics [28]. Recent developments and challenges in complex question answering have been exhaustingly surveyed in [24]. This research examines the methods available in answering complex questions in the context of the knowledge base.

3.2. Text-based

Text-based systems answer questions from existing documents during the two main operations: passage retrieval and machine reading comprehension [10]. Passage retrieval task extracts the k-top relevant documents to the question by comparing the question and the document vectors using a distance measure. Passage retrieval is a branch of information retrieval that reduces the search space to extract answer. There are many researches in this area, trying to improve the retrieval models, so that the best candidates can be extracted to help find the answer.

Initial works used term-based methods as lexical adaptations of TF-IDF and BM25 [11], [12]. In these methods, retrieval is based on the bag of words concept, and the ranking function is calculated based on the term and inverse document frequencies. To address the challenge of sparse vectors with high dimensions, considering the semantic property for embedding vectors through latent semantic analysis and the

concept of dense retrieval [13], [14], [29] can improve the performance. Recently, efforts have been made to consider a small set of question-answer pairs to create vectors via dual encoders [14], [15]. Also, the inner product ranking function between the question and the passage vectors is used [15], [30]. An overview of text-based methods from the perspective of information retrieval and deep learning is presented in [31].

On the other hand, the processing of complex questions has been considered in many text-based systems [10], [14], [16], [32], primarily when the answer to the question cannot be extracted with a single text piece or when multi-step inference must be performed on several text pieces to find the answer. Such questions are called multi-hop. Concerning multi-hop questions, scalability and the ability to extract the path to the answer are two essential factors.

Derived from the sequential nature of multi-hop question answering, MDR [15] uses an iterative process and maximum inner product search technique along with dense retrieval [14] to speed up the extraction of a sequence of passages from a large pool of documents. In the first step, the most similar passage to the question is extracted, and in the following steps, a new question is generated by combining the answers of the previous steps and the initial question. The newly generated question is used to compare and find the similar passages in each step.

Although, the present research to answer the question is not just text-based and belongs to the hybrid category, it employs the MDR method to solve the challenges of scalability, interpretability, and the ability to extract answers from the text.

3.3. Hybrid

Although a large volume of information is in the text format, the extraction of an answer from a text has a lot of complexities, due to the diversity of manners of expressing information in natural languages. On the other hand, in a KB, information is expressed in specific structures that make it easy to extract. However, even the largest KBs suffer from information coverage problem. Therefore, it is evident that these two sources of information can complement each other regarding the coverage and simplicity of information extraction.

In general, systems that take the advantages of these two areas operate in two main models: early fusion and late fusion [2]. In the early fusion, both sources are searched simultaneously, while in the late fusion, each source is searched separately, and later the answers are fused. The former has been shown to perform better than the latter.

Some of the models presented in this category are either primarily text-based systems, extended to be able to use a KB [33] or, vice versa [34], [35]. ODQA [36] converts all available sources to text and then uses retrieval and reading tools. A few researches have extracted an answer by simultaneously using text and a KB [2], [3]. These systems do not support multi-hop questions.

The first attempt to simultaneously extract an answer from text and a KB was to use key-value memory alongside universal schema [37], though not considering the rich relationships between facts and textual parts. Another approach, GRAFT-Nets [2], retrieves a response-related subgraph by creating heterogeneous graphs of entities, facts, and text pieces instead of randomly extracting them. However, the generated graphs are often huge and not scalable.

By providing a way of learning to develop nodes, PullNet [3] has shown that it can gradually produce graphs, resulting in smaller graphs. PullNet assumes that the answer can be extracted if it exists in the KB form. In addition, the explanation of how to find the answer, which is necessary in the real world [17], [18], cannot be extracted. Another problem is that PullNet does not generate optimal graphs, because it creates the initial graph with all the question entities and does not consider the sequential nature of the question requiring multi-step inference. In other words, PullNet ignores the relationships between question entities.

As the information containing the answer to the multi-hop question cannot be obtained in a single shot [15], the present research believes that not all the question words should be considered simultaneously, but should be added step by step in the search.

The system presented in the current research falls into the category of hybrid systems. This research aims to present a system that, like PullNet, processes multi-hop questions with the help of structured and unstructured sources. Moreover, it has the ability to scale up, extract response explanation, and extract the final answer from both sources, in such a way that it can control the search space to increase the efficiency of the system.

3.4. Explainable Question Answering System

In the past, in all areas, systems focused only on extracting the correct answer. Recently, in addition to extracting the correct answer, the inference method has also been considered [17]. Most AI systems are

black boxed, meaning that the internal process is hidden from users. The explanation will answer the question of how the model came to a particular conclusion starting from a problem [38]. Therefore, the presence of explainable artificial intelligence will lead to system transparency and thus trust in the system. According to previous research, the nature of interpretations in various AI systems varies depending on the field and scenario [18]. Machine reading comprehension (MRC) systems have recently turned their attention to extracting explanations [38]. The purpose of these systems is to extract the answers to natural language questions on a corpus. Extraction of explanation for MRC systems is presented in two methods of extractive and abstractive. The first method involves extracting supporting paragraphs to achieve the answer [15], [39]. But the second method achieves the final answer by inferring on the summary of several pieces of passages [40]. In this research, an MRC system [15] has been used to solve the challenge of past hybrid systems, which uses the extraction method to interpret the response. Therefore, the proposed hybrid method interprets the answer using a number of supporting texts and triples.

4. Background

In the proposed system, to solve the challenge of state-of-the-art hybrid systems, the combination and development of two systems has been used. The first system is called PullNet [3], which is a hybrid system and the next system is called MDR [15], which is text-based. Therefore, in this section, the methods of these two systems are presented.

4.1. PullNet system

The working method of the PullNet system is shown in Figure 2. The PullNet system uses both a text corpus and a KB to build a graph. This graph expands repetitively. Initially, nodes V_e are constructed using all the question entities. Next, the KB triples (V_t) and passages (V_d) in text, which contain the existence nodes V_e, are added to the graph (Refer to the PE area in Figure 2.). The new entities in the existence nodes V_d and V_t in PI area is then added as the new node (Notice the bold nodes V_e in the EI area in Figure 2.). For graph expansion in next iteration the entity nodes in the EI area of each iteration are used (Notice the nodes shown within the dashed circle in Figure 2). The PullNet system for graph expansion and finding the final answer is based on training through a pair of questions and answers. The final answer is in the form of an entity in the knowledge base in the last iteration (Notice the nodes shown inside the circle in the n-*th* iteration in Figure 2.).

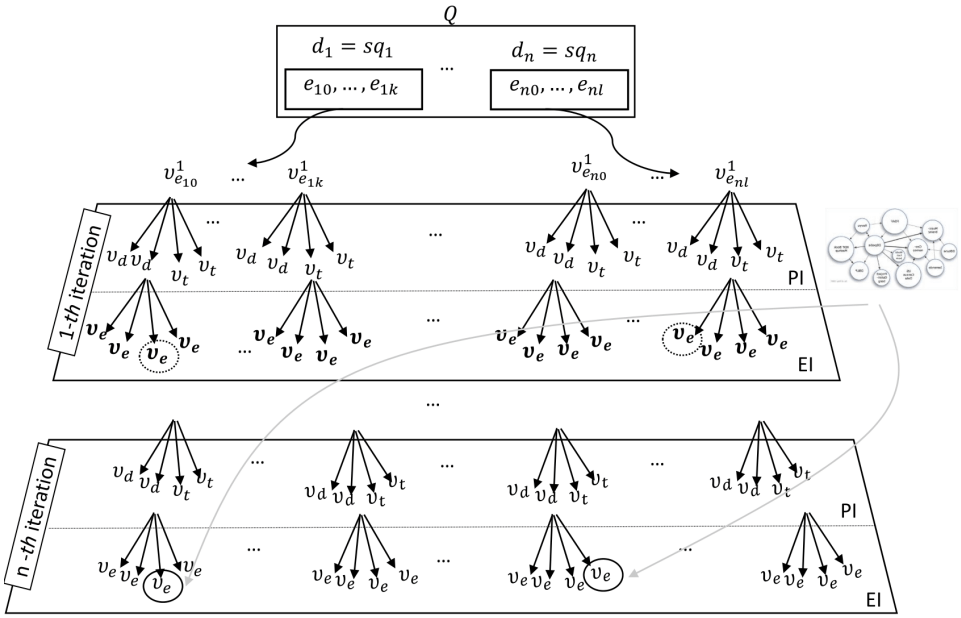

Fig. 2. The graph expansion process in the PullNet model.

4.2. MDR system

The MDR system receives a multi-hop question, q, to retrieve the sequence of texts, $P_{seq}: \{p_1, \dots, p_n\}$, to achieve the final answer. The desired sequence provides information to reach the final answer. The retrieval finds a set of top-scoring candidate sequences, $\{P_{seq}^1, \dots, P_{seq}^k\}$. Due to the inherent nature of multi-step queries, the MDR system uses a repetitive method and models the probability of finding sequences of texts as Eqs. (1) and (2).

$$\mathrm{P}(P_{seq}|q) = \prod_{t=1}^{n} P(p_t|q, p_1, \dots, p_{t-1}) \quad (1)$$

$$P(p_t|sq_t, p_1, \dots, p_{t-1}) = \frac{\exp(\langle p_t, q_t \rangle)}{\sum_{s \in (v_e \cup v_d \in G^t)} \exp(\langle \boldsymbol{s}, \boldsymbol{sq_t} \rangle)}$$

where $q_t = g(q, p_1, \dots, p_{t-1})$ and $p_t = h(p_t)$ (2)

In each step, a new question is created using the answers obtained from the previous steps. Two functions h(.) and g(.) are encoders to produce the dense representation of intermediate answer.

5. Approach

As mentioned, the current research belongs to the complex (often called multi-hop) question-answering systems, trying to extract answers from both text and KB sources. It is based on three fundamental axes, described as follows. The motivation for using these three axes is given in Section 7.3.

(i) A complex question can be defined as a question that has several relations, entities and may contain various constraints, e.g., temporal, spatial, aggregation, ordinal, etc. According to this definition, in most cases, a complex question can be converted into several minor questions (sub-questions) that should be executed in a specific order to obtain the correct answer. Decomposition of the question into sub-questions has led to the improvement of system performance.

(ii) The objective is to find answers using both entity-linked text and a KB source. Thus, an effective way to obtain the answer to a given question is a graph, as it can clearly show the relations among passages, entities, and RDF triples. The current work constructs the proposed approach based on the PullNet [3] method, extending the graph in several steps.

(iii) The answer to multi-hop questions, usually can be searched in a sequence by finding pieces of information. In this case, the proposed approach is built on the dense retrieval multi-hop system, MDR [15], which attempts to extract the best sequence from a pool of documents. However, in our approach, the pool containing passages and facts is searched.

The present study provides a solution by integrating the three considered axes to balance the accuracy and the efficiency, while extracting the answer from both sources unlike the hybrid state-of-the-art system, PullNet.

Throughout the article, triple and fact are used interchangeably. In addition, the concept of the document and the passage are the same and are considered as a single sentence.

5.1. Model

The architecture of the proposed model, GraphMDR[1], is based on four modules. This architecture is shown in Figure 1. Figure 2 illustrates the structure of the proposed system using a sample question for better presentation. The parts in the gray box marked with a dashed line in Figure 2 are iteratively processed. At each stage of iteration, Question Graph Expansion (QGE) module and Sequence Retrieval (SR) module are examined for one of the sub-questions of Sub-Questions Generation (SG) module. These modules are described in some details, here.

Question Decomposition (QD):

First, for a given complex question, Q, containing several relationships, entities, and constraints, the sequence of sub-questions based on the order of execution, $Q: \{sq_1, sq_2, \dots sq_n\}$ are generated. As GraphMDR is a hybrid method, it utilizes an open domain method [10][2] for question decomposition. The input query is considered as $S = [w_1, \dots, w_n]$, containing n words. Using a small set of annotated questions, the model is trained to map the input question to c points $ind_1, \dots, ind_c$. The trained model encodes the input question by BERT as in Eq. (3).

$$U = BERT(S) \in R^{n \times h} \quad (3)$$

The probability $P = (i = ind_j) = Y_{ij}$ Specifies the probability that the i-*th* word is the j-*th* generated index. The model extracts a number of points that lead to the highest continuous probability as Eq. (4).

$$ind_1, \dots, ind_c = argmax\, P(ij = ind_j) \quad (4)$$

The analysis method is performed in such a way that, with little supervision dataset, three types of inferences, namely intersection, bridging, and comparison, (as described in [1]) are identified. These inference types are obtained according to the frequency of types available in a set of 400 questions.

[1] https://github.com/SAsadifar/GraphMDR

[2] https://github.com/shmsw25/DecompRC

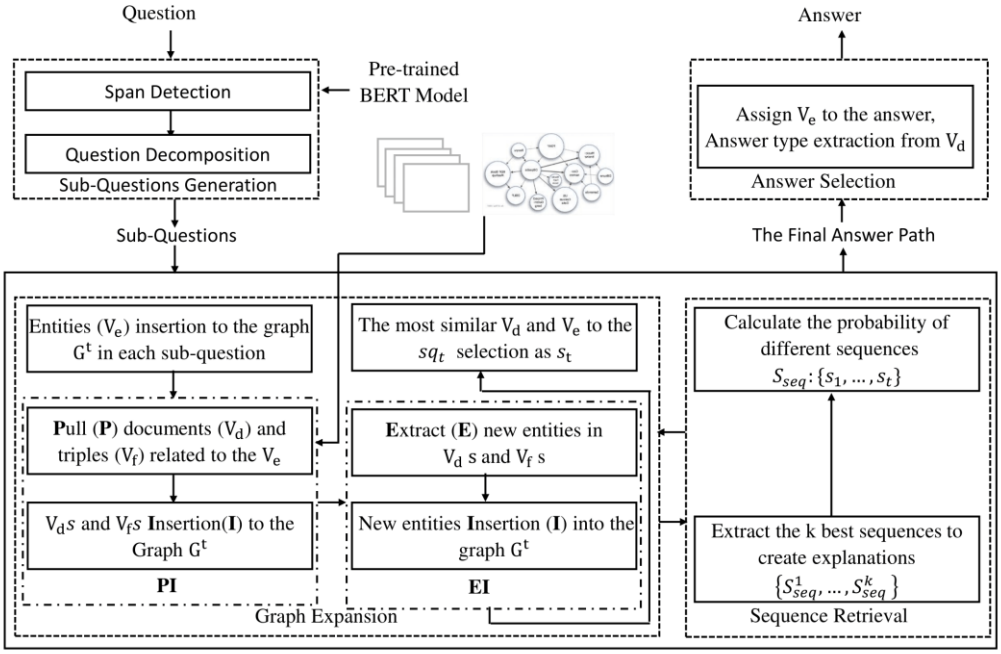

Fig. 3. The architecture of the proposed model, GraphMDR.

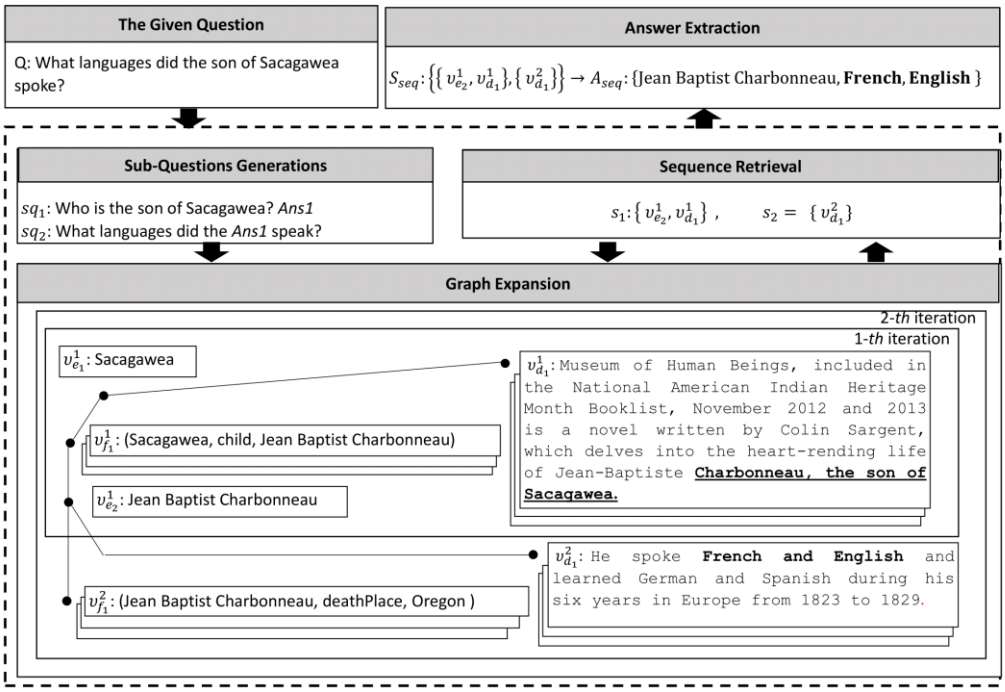

Fig. 4. The architecture of the proposed model, GraphMDR using a sample question.

5.1.1. Graph Expansion (GE)

To find the answer in a massive pool of passages and knowledge base triples, the current work employs a heterogeneous graph representation model utilized in previous works, PullNet [3] and GRAFT-Nets [2]. We define the graph, G as $G = \{V, E\}$, where nodes (V) are of one of the types: entity nodes related to the question (V_e), document text nodes (V_d) and fact nodes containing those entities (V_t). The edges, E, show the connection between each V_e and V_d or V_t that contain the entity V_e.

In this research, each document d is a sentence of words $(w_1, \dots, w_{|d|})$. It is assumed that an entity linking system is executed on the sentences before they are indexed [41]. The TAGME[3] system is one of the most common systems for identifying entities in the text, which has shown good performance in past researches [42], [43]. In the proposed method, the TAGME system is used to link the mentions in the text to Wikipedia entities. The result is $\mathcal{L}$, a set of links (v, d_p) that represent the link of the entity v to the position p in the document d.

5.1.1.1. Graph expansion process

The proposed model for graph expansion process consists of three categories of operations: (i) intra-iteration, (ii) intermediate, and (iii) the final answer selection.

Intra-iteration operations include the following:

(i) Add existing new entities in sub-question $\mathrm{sq_t}$ to graph $\mathrm{G^t}$ as a new υ_es. If no entity is detected, $\mathrm{sq_t}$ is added as a υ_d.

(ii) **P**ull (P) top-related sentences and triples related to the $\mathrm{V_e}$s and **E**nter (E) new nodes ($\mathrm{V_d}$ or $\mathrm{V_t}$) to the graph $\mathrm{G^t}$. Then Create graph edges between each $\mathrm{V_e}$ and the associated $\mathrm{V_d}$ or the $\mathrm{V_t}$ in the graph $\mathrm{G^t}$. This operation is called PE for short (See Figure 5). This step run in parallel on the sources, including sentences and KB triples, each of which consists of two minor steps: (1) sentences (using a set of entity links $\mathcal{L}$) and KB triples which include $\mathrm{V_e}$s, and (2) from the selected sentences and triples of the previous minor step, the most similar to the current query in the vector space are selected and added to the graph. If no entity is existed, we run only minor step 2 from all the sentences and KB triples.

(iii) **E**xtract (E) existing new entities in $\mathrm{V_d}$ s and $\mathrm{V_t}$ s from the PI operation and **I**nsert (I) them into the graph $\mathrm{G^t}$. This operation is called EI for short. (See Figure 5)

Intermediate operation include operation between two iterations. This operation used to select the most similar $\mathrm{V_d}$ and $\mathrm{V_e}$ nodes in the previous iteration to the current query, which will participate in the PI operation in the next iteration for graph expansion. The selected nodes are considered as intermediate responses which shown in a circle with thick line in Figure 5.

The final answer selection operation involves selecting the most similar $\mathrm{V_d}$ and $\mathrm{V_e}$ nodes (in all iterations of graph) to the question generated in the last iteration.

5.1.1.2. Disjunctive and Conjunctive Sub-questions

If there is a disjunction or conjunction between two or more sub-questions, the entities of the sub-questions are added to the graph in one step. The search for similar passages and triples for each entity added to the graph is based on a sub-question containing that entity. The example of Question **q3** is one of the multi-hop questions containing conjunction in Table 10 of Appendix A.2.

5.1.1.3. GraphMDR vs. state-of-the-art hybrid QA, PullNet

According to Figure 2 and 5, the proposed model for graph expansion is built upon PullNet, with main differences to increase the efficiency:

1. *Prevent graph widening*: by considering the Figure 5, in the proposed system, instead of all the entities in the given question in PullNet system, only the entities of sub-question $\mathrm{sq_1}$ are entered in the graph $\mathrm{G^1}$ in the first iteration. In each iteration, *t*, in addition to extending the graph through the entities available in the graph G^{t-1}, we add the entities of sub-question sq_t to this graph (Refer to the item 1 in the graph expansion process in the *t-th* iteration). Gradually entering entities into the graph, results in smaller graphs and nodes closer to the response than the PullNet (See Figure 5).

[3] https://tagme.d4science.org/tegme/

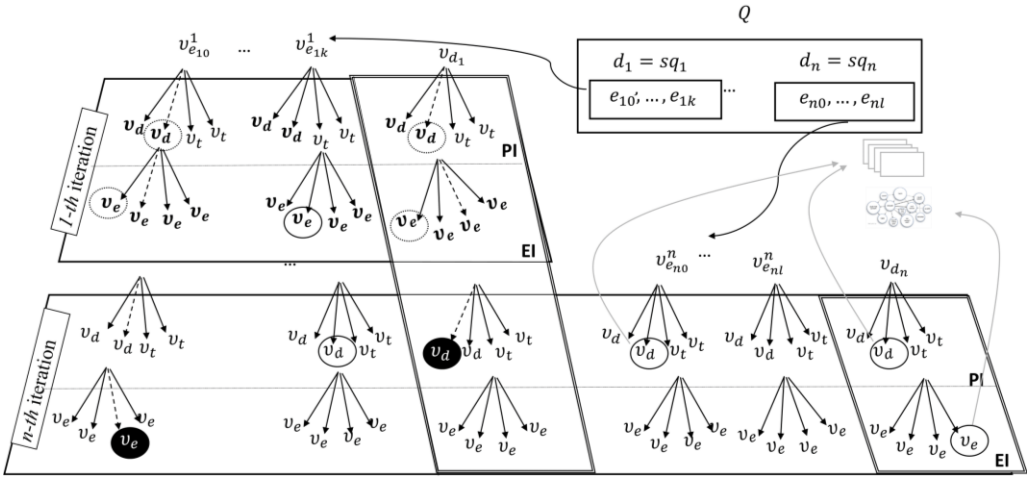

Fig. 5. The graph expansion process in the proposed model, GraphMDR

2. *KB independent model*: If no entity is available in each sub question sq, the sq is considered as υ_d. For example, we draw the reader's attention to the example of q_3 from Table 9 in the Appendix A.1 section. The generated sub-question sq_1 does not contain an entity, so the whole query is added as a υ_d node in the graph. Unlike PullNet, the proposed method, in addition to entities, also supports (sub-) questions without entities. Therefore, it has the ability to answer (sub-) questions that are not related to the knowledge base. The parallelograms shown by the two lines in Figure 5 are considered in the absence of an entity at that stage.
3. *Use of advanced language models*: We need to compare the three sections to find the entity, the sentence and the triple candidates: (i) adding sentences and triple nodes (υ_d, υ_t) to the graph (Refer to the PI area in Figure 5), (ii) candidate nodes recognition to expansion in next iteration (Refer to the EI area in Figure 5), and (iii) final answer recognition in last iteration. Unlike PullNet, in the proposed model, instead of a method based on sparse vectors, vectors in space model are used. We retrieve top-most vectors similar to the embedding vector of current question $\boldsymbol{q}$ in each iteration. Maximum inner product search is used to calculate similarity.
4. *Create current question based on previous answers*: In some questions, the answer of the current sub-question depends on the answer of the previous sub-questions. Thus, not capturing the answer of the previous steps reduces the accuracy of the final answer. By entering the sub-question sq_t, in each iteration, its dense vector is generated, $\boldsymbol{sq_t}$. The answers in the previous steps are $(s_1, \dots, s_{t-1})$, including sentences and entities. The following section (§ 5.1.3) explains the details of using this method. The query at each iteration involves concatenating the embedding vector of answers from the previous sub-questions, $(\boldsymbol{s_1}, \dots, \boldsymbol{s_{t-1}})$, and the current sub-question $\boldsymbol{sq_t}$ as defined in Eq. (5).

$$\boldsymbol{q} = E_p(sq_t, s_1, \dots, s_{t-1}) = \boldsymbol{sq_t} \oplus \boldsymbol{S_1} \oplus \dots \oplus \boldsymbol{S_{t-1}} \quad (5)$$

Example $\mathbf{q_2}$ and $\mathbf{q_3}$ in Appendix A.1 in Table 9 are provided to clarify the issue.

The question $\mathbf{q_2}$ is divided into two sub-questions $\mathbf{sq_1}$: "Which film is based on an opera by Giacomo Puccini?" and $\mathbf{sq_2}$: "In what city did the \"Prince of tenors\" star in a film?". The extracted passage as the answer to the first question is "It is based on the 1900 opera Tosca by Giacomo Puccini, which was adapted from the 1887 play by Victorien Sardou." From "Tosca (1956 film)" Wikipedia page.

The second sub-question alone does not elicit an answer "It was made at Cinecittà in Rome", because the answer does not depend on the named entity "Prince of tenors" of the current sub-question; while it is related to the answer of the previous step. As a result, the sub-question can only be answered correctly if combined with the answer of the previous steps. As another example is the second sub-question $\mathbf{sq_2}$: "the university is located in what city?" from $\mathbf{q_3}$. The answer "New York City" can only be obtained if $\mathbf{sq_2}$ is combined with the answer of the previous step, "Ralph Franklin Hefferline (15 February 1910 in Muncie, Indiana – 16 March 1974) was a psychology professor at Columbia University".

The answer can be extracted according to the passage **sp**$_1$: "Columbia University (also known as Columbia, and officially as Columbia University in the City of New York) is a private Ivy League research university in New York City.". From "Columbia University" Wikipedia page and the triple **sf**$_1$:" (dbr:Columbia_ University, dbp:city, dbr:New_York_City)" from the resource "dbr:Columbia_University" in DBpedia.

5. *Use of advanced language models*: when adding passages (v_d) to the graph (refer to the item 3 in the graph expansion process and PI area in Figure 5 in the *t-th* iteration), instead of a method based on sparse vectors, dense passage and the query vectors are compared.
   The query and passage encoder can be implemented using any neural network. Here, we use two independent neural networks. We retrieve topmost vectors similar to the embedding vector of current question $\boldsymbol{q}$ in each iteration. Maximum inner product search is used to calculate similarity.
6. *Considering all types of hops*: in the PullNet system, only the entity nodes in the EI area of each iteration are used for expansion in the next iteration. Figure 2 shows the extensible nodes in bold ($v_e s$) for PullNet system. It is possible that the middle or the final response is in a sentence similar to the existing sentence nodes. Ignoring sentence nodes for later expansion (sentence-sentence hops) reduces response accuracy. In the proposed method, in addition to the entity nodes, the sentence nodes are also used in each iteration for expansion in the next iteration. Figure 5 shows the extensible nodes in bold ($v_e s$ *and* $v_d s$) for GraphMDR model. The most extensible nodes to the current query are selected for the next iteration (shown in Figure 5 with the dashed circle box).
7. *KB-independent response*: the PullNet system for finding the final answer is based on training through a pair of questions and answers. The final answer is only in the form of an entity in the knowledge base (Notice the nodes inside the circle in the Figure 2). In the proposed method, the final answer is selected from all nodes of the entity and the sentence in the graph. The choice of answer is based on the similarity between the candidate nodes with the question created in the last iteration.

5.1.1. Sequence Retrieval (SR)

There are two important objectives for extracting sequences from sources containing intermediate answers: i) to create explanations for the final answer and ii) to possibly provide the final answer from the text pieces (unlike previous works, in which extracting the answer as an entity was only possible). To achieve mentioned goals, we follow the concepts of the method introduced by [15].

In [15], the problem of finding the answer to a multi-hop question has been transformed to the problem of finding the best sequence $P_{seq}: \{p_1, \dots, p_n\}$ of intermediate answers among the k identified sequences $\{P_{seq}^1, \dots, P_{seq}^k\}$ .These sequences only contain sentences related to the question. The probability of finding a sequence is defined as:

$$\mathrm{P}(P_{seq}|\mathrm{q}) = \prod_{t=1}^{n} P(p_t|q, p_1, \dots, p_{t-1}) \quad (6)$$

$$P(s_{t_i}|sq_t, s_1, \dots, s_{t-1}) = \frac{\exp(\langle \boldsymbol{s_{t_i}}, \boldsymbol{sq_t} \rangle)}{\sum_{s \in (v_e \cup v_d \in G^t)} \exp(\langle \boldsymbol{s}, \boldsymbol{sq_t} \rangle)}$$

where $sq_t = g(sq_t, s_1, \dots, s_{t-1})$ and $s_{t_i} = h(s_{t_i})$ (7)

Here, we employ the key idea that multi-hop questions are often sequences of sources to extract the final answer. The SR module examines a set of sub-questions, $Q: \{sq_1, sq_2, \dots sq_n\}$, generated in the SG module, as well as the graph G^t in the *t-th* iteration, generated with the GE module. Since the entities in the fact nodes are presented in the graph as entity nodes, the present study chooses the intermediate answers from $v \in v_e \cup v_d$.

The SR module needs to retrieve sequence $S_{seq}: \{s_1, \dots, s_n\}$, where $s_t: \{s_{t_1}, \dots, s_{t_r}\}$ represents a set of nodes ($v \in v_e \cup v_d$) in the graph G^t that are candidate source snippets for intermediate answers in the *t-th* iteration. In addition, the k best sequences, $\{S_{seq}^1, \dots, S_{seq}^k\}$, from several candidates are extracted. The probability of selecting a node ($v_e \cup v_d$) is modeled as follows, which ultimately results in the probability of sequence S_{seq} as Eqs. (8) and (9).

$$\mathrm{P}(S_{seq}|\mathrm{q}) = \prod_{t=1}^{n} \prod_{i=1}^{r} \mathrm{P}(s_{t_i}|sq_t, s_1, \dots, s_{t-1}) \quad (8)$$

$$P(s_{t_i}|sq_t, s_1, \dots, s_{t-1}) = \frac{\exp(\langle s_{t_i}, sq_t \rangle)}{\sum_{s \in (v_e \cup v_d \in G^t)} \exp(\langle s, sq_t \rangle)} \quad (9)$$

In the *t-th* iteration, the maximum inner product search is performed on the dense representation of all the nodes of the entity (V_e) and the sentence (V_d) in the graph G^t. The operator, $\langle ., . \rangle$, is defined as the inner product between vectors $v \in v_e \cup v_d$ and sq_t vector at each iteration.

where $sq_t = g(sq_t, s_1, \dots, s_{t-1})$ and $s_{t_i} = h(s_{t_i})$, and two functions h(.) and g(.) are encoders that produce the dense representation of intermediate answer $\boldsymbol{s_{t_i}}$ and sub-question $\boldsymbol{sq_t}$, respectively.

In each iteration, $\boldsymbol{sq_t}$ and the previous set of intermediate answer sources are concatenated, and vector $\boldsymbol{q_t}$ is obtained by the g(.) encoder. The proposed approach for extracting sequences from source snippets

containing intermediate answers is similar to the method mention in [15], as both utilize dense retrieval. However, there are three differences: (i) for t = 1, the present study relies on sub-question sq_1 instead of the whole question; (ii) at the *t-th* iteration, instead of considering the representation of the question, sub-question sq_t participates in the question representation process; and (iii) in the proposed method's *t-th* iteration, several candidate sources are selected to obtain an intermediate response from the entity (V_e) and the sentences (V_d) nodes in graph G^t with the help of entities in previous iterations.

The evaluation in Section 6.2 shows that significant improvements are gained by considering these modifications.

5.1.2. Answer Extraction (AE)

The final answer selection operation involves selecting the most similar V_d and V_e nodes in all iterations of graph to the question generated in the last iteration, $sq_n = g(sq_n, s_1, \dots, s_{n-1})$.

Simple heuristics for answer extraction are employed. We consider s_n in the selected S_{seq} sequence in the *n-th* iteration. For each s_{n_i}, if s_{n_i} is an entity node, it is the answer. Otherwise, the sub-question sq_t is searched for the expected answer type and, based on the answer type, the answer is extracted.

5.2. Model Training

The proposed model includes two different types of training:

The first training is for identifying the best entities used for expansion at each iteration. The current study follows the PullNet [3] method, in which the learning is achieved by considering the question-answer pairs for finding the shortest paths between the question entities and the answer entity.

The second training tries to identify the best sequences of entities and sentences to answer the question. It follows the MDR [15] method, with the difference that we include several positive and negative KB facts in the training process, in addition to considering the question along with the related positive and negative sentences.

6. Experiments and Results

This section describes multi-hop question-answer datasets and baselines and reports the results of a comparison between the proposed approach and the baseline systems.

6.1. Databases and Baselines

There are two main categories: simple questions (single-hop) and complex questions (multi-hop). For simple questions, there are KB-based databases, such as Wikimovie [44], and text-based databases, such as TriviaQA [45] and Squad [46]. For complex KB-based questions (multi-hop), WEBQUESTIONSSP [47], WebQuestions [35], COMPLEXWEBQUESTIONS [1], and MetaQA [48] datasets, and for text-based questions, HOTPOTQA [39] dataset have been used in the literature.

The proposed method aims to address some challenges of the PullNet methods [3] and the multi-hop dense retrieval MDR method [15]. The PullNet approach answers complex questions using a KB and text, while MDR answers complex questions using a pool of text sentences. Therefore, we evaluate GraphMDR with PullNet [3] using MetaQA [48], WEBQUESTIONSSP [47], and COMPLEXWEBQUESTIONS [1] databases and with MDR method [15] using HOTPOTQA [39] database.

MetaQA [48]**:** This database is based on WikiMovies [44], and uses some Wikipedia texts to help answer questions. Also, up to 3-hop questions added to the previous single-hop collection, known as the Vanilla version. The KB includes 43k entities and 13k triples.

The questions have been transformed into 2-hop and 3-hop questions, based on some patterns, in a sequence of 2 and 3 simple questions, respectively. Therefore, the basis for constructing these questions is the combination of sub-questions. Since the present approach is based on the decomposition of questions into simple questions, this dataset is entirely consistent with the proposed method. Table 8 in Appendix A.1.2 shows examples of 2-hop and 3-hop questions, along with the type of each question and its sub-questions.

WEBQUESTIONSSP [47]: This database is based on the WebQuestions database [35], where 84% of the questions are simple, and the rest are up to 2-hops questions that can be answered using Freebase alone. On the other hand, Wikipedia texts have also been used as a text corpus to provide a composite platform that requires text and a KB to respond.

COMPLEXWEBQUESTIONS [1]: This database has created more complex questions by adding constraints and expanding the WEBQUESTIONSSP database [47] entities.

HOTPOTQA [39]: This database has up to 2-hop questions, is based on Wikipedia, and provides the possibility of answer interpretation by having a set of supporting passages to reach the answer.

Because GraphMDR has the ability to search using both textual and KB sources, in order to make it possible to evaluate, the *k* facts of KB most similar to the supporting passages in the HOTPOTQA's database are selected using the BERT embedding vector comparison method [49], and these are used alongside the supporting passages. In addition, since GraphMDR is based on decomposing questions into sub-questions, the data set questions are decomposed. Then, the existing supporting passages for each question are separated based on the sub-questions. Examples of HOTPOTQA dataset enriched with supporting triples are shown in Table 9 of Appendix A.1.1 section.

The questions in the HOTPOTQA dataset have different numbers of supporting passages. For example, question $\mathbf{q_1}$ from Table 9 in Appendix A.1 section, has three supporting passages. Since the proposed method is based on question decomposition, the three supporting passages are separated based on the generated sub-questions. Then, for each supported passage, supported triples are generated. As can be seen, the first sub-question $\mathbf{sq_1}$ contains one supporting passage, and the second sub-question $\mathbf{sq_2}$ contains two supporting passages.

Table 2 provides data statistics on the training, Validation (dev), and test categories of the databases used for the current paper's evaluation.

Table 2: Statistics of databases used in the evaluations

Benchmark	Training	Validation	Test
MetaQA	329282	39138	39093
WEBQUESTIONSSP	2848	250	1639
COMPLEXWEBQUESTIONS	27623	3518	3531
HOTPOTQA	90564	7405	7405

6.2. Evaluations

As mentioned in the previous section, the two basic systems in the current work are PullNet [3] and MDR [15], of which MDR can provide responses only from text snippets, and PullNet is able to extract responses only as an entity from a KB. The proposed system, GraphMDR, is compared separately with two base systems, PullNet (see Tables 3, 4 and 5) and MDR (see Table 6 and Fig. 6). One of the challenges addressed in the proposed hybrid system, GraphMDR, is the possibility of extracting the response in the textual source or in the form of an entity. The HOTPOTQA dataset contains questions, some of which are in the text and some in the form of an entity. Therefore, the proposed method and the hybrid PullNet method as well as MDR are evaluated simultaneously on the HOTPOTQA dataset (see Table 7).

All experiments are conducted on a machine with a 4 Core Intel Xeon E5 CPU @ 2.00GHz with 16 GB of RAM. The FAISS[4] library is used to store dense vectors and calculate the best candidates. This open-source library is very effective for searching for similarities between dense vectors, because searching through vector clustering allows the system to search through the billions of dense vectors quickly. Answer selection is made using the Transformer[5] framework. To identify the answer, the ELECTRA model is used, which is reported in [15] to have the best performance. In the evaluations, whenever the difference in results is small, a *t-test* is also performed by SPSS v11.0. Results with a statistically significant difference are highlighted in the tables. .

Table 3 shows the accuracy of GraphMDR and PullNet for the three provided KB-based databases. Similar to PullNet, GraphMDR provides a combination of text and a KB to answer questions in all three databases. As seen in the previous work [3], 50% of the knowledge base was used to create a combined platform. In addition to 50% of the KB, text corpus was also utilized. Text corpus uses TAGME entity link tool to connect to the KB. Table 3's evaluation metric is Hits@1, which indicates the accuracy of the answer with the highest prediction. Similar to previous work [50], BERT-based vector space comparisons have been used to evaluate the extraction of the final response based on Hits@1 for string similarity.

Table 3: Comparison of PullNet and GraphMDR systems based on Hits@1 metric

	MetaQA (test)			WEBQUESTIONSSP (test)	COMPLEXWEBQUESTIONS (dev)
	(1-hop)	(2-hop)	(3-hop)		
PullNet	92.4	90.4	85.2	51.9	33.7
GraphMDR	92.4	90.4	86.1	**71.9**	**62.3**

[4] https://gitHub.com/facebookresearch/faiss

[5] https://huggingface.co/transformers/

Table 5: Comparison of PullNet and GraphMDR based on the amount of entities/ recall of multi-hop

	MetaQA (3-hop) (test)	COMPLEXWEBQUESTIONS (test)
PullNet	63.3/0.98	44.1/0.68
GraphMDR (using sq)	**34.3/0.98**	**19.5**/0.78
GraphMDR (using q)	62.3/0.98	40.5/**0.81**

As shown in Table 3, GraphMDR is more accurate than PullNet. The higher accuracy can be interpreted in two main factors. The first one is that in the proposed method, passage retrieval is done based on embedding vectors, while in PullNet, traditional methods (TF-IDF) are used. The second factor is the involvement of the previous steps results in retrieving the pieces of information containing the answer in each step.

The accuracy of both systems in the MetaQA dataset is much higher than the two other datasets, WEBQUESTIONSSP and COMPLEXWEBQUESTIONS, because in MetaQA dataset, the questions are based on specific and limited patterns. As can be seen, GraphMDR has a better result for 3-hop MetaQA dataset and has presented similar results for 1-hop and 2-hop.

Table 4 provides comparison of GraphMDR and PullNet results using Google snippets without entity links to a KB, Wikipedia text data with existing KB links, and KB data alone for multi-hop COMPLEXWEBQUESTIONS database.

Unlike PullNet, the GraphMDR method has the ability to extract responses from text snippets. As a result, as shown in Table 4, GraphMDR provides more accuracy for Google snippets and Wikipedia than PullNet. This result is especially in the case of Wikipedia, because the entities in the text are linked to a KB. In addition, the GraphMDR works based on embedding vector space, so it is more accurate on Hits@1 metric. In the case of Freebase, the accuracy of both methods are the same, because only KB entities are used to extract the response.

Table 4: Comparison of PullNet and GraphMDR based on Hits@1 metric using COMPLEXWEBQUESTIONS (test) database

	Google snippets	Wikipedia	Freebase
PullNet	29.7	13.8	45.9
GraphMDR	**45.2**	**52.2**	45.9

To analyze the efficiency-accuracy trade-off, the number of entities and the recall of the two systems, PullNet and GraphMDR, using a full KB are compared. As shown in Table 5 despite the production of a smaller graph by GraphMDR, it has also reported more accuracy.

In GraphMDR, the number of iterative steps to expand the graph and retrieve the sequence of answers is equal to the number of sub-questions (GraphMDR (using sq) in Table 5). It should be noted that, as stated in the explanation of the question graph expansion method in Section 5.1.2.2, sub-questions that are connected by disjunction or conjunction, are considered as a step.

We were interested in determining the effect of using query decomposition on the proposed method. Therefore, the results of the proposed method are reported in two experiments with (GraphMDR (using sq)) and without the use of sub-questions (GraphMDR (using q)) along with the results of PullNet (Table 5).

As can be seen, when the question is divided into sub-questions, the number of entities within the graph is significantly reduced compared to the other two cases. This observation is logical, because the graph expansion is done with the guidance of sub-questions (similar to the way the human mind works). In the other two cases, the whole question is used from the beginning of the process to extract the answer, which lead to the production of a larger graph.

In the second case (GraphMDR (using q)) all the question words are considered at once for graph construction, as in the PullNet method. Therfore the number of graph entities is much higher than the previous case (GraphMDR (using sq)), while the accuracy does not increase significantly. The increase in accuracy compared to the PullNet method is due to the use of 1) dense retrieval approach, and 2) the results of the previous steps, discussed in the description of Table 3.

Table 6 compares the retrieval efficiency of related passages or facts in both GraphMDR and MDR systems. As seen, GraphMDR provides better results than MDR. This improvement in results can be explained as follows. GraphMDR, retrieves nodes in the graph that are related to the entities, existing in the sub-question being processed. The current sub-question is used for comparison, so the results are closer to the supporting facts and passages in the database. Supporting

facts and passages are created based on the sequential nature of answering multi-hop questions.

In Figure 6, the results from running two systems, MDR and GraphMDR, based on different *k* inputs on the HOTPOTQA database is shown (It should be noted that the MDR system was reimplemented).

Since HOTPOTQA questions are up to 2-hops, GraphMDR contains a maximum of two sub-questions that include two iterations to find the answer. Consequently, there is a maximum sequence of two sets. In each set, the most appropriate doc or entity nodes are retrieved. *k* is considered as the total number of nodes suitable for recovery in sequence in a maximum of two iterations ($\frac{k}{2}$ in each iteration).

As shown in Figure 6, for each input, *k*, GraphMDR offers higher accuracy in shorter execution time. These results can be interpreted in two ways: First, according to Background Section, MDR uses all the question words to find the sequence of texts containing the answer. In contrast, GraphMDR uses sub-question order sequences to find text snippets or entities similar to the current sub-question. This fact leads to more accurate results as well as faster speed.

Secondly, MDR only searches for answers using a sequence of text snippets. In contrast, the answer is extracted more accurately and quickly in a structured data if an answer is available in a KB. GraphMDR achieves better results by using this feature.

In the current research, in addition to extracting the final answer, the method of inferring the answer or explanation is also considered. Therefore, in this section, we have used the logical rigor concept used in MRC [51] to evaluate the proposed model. This concept includes two criteria of exact match (EM) and F1 measure. EM means the exact match of the extracted string with the desired string. Ans is used to evaluate the model in extracting the final answer and Sup is used to evaluate the model in extracting the path to reach the answer. Joint has been used to evaluate the model in simultaneously extracting the final answer and the path to the answer.

Table 6: Evaluation based on retrieval performance in recall at k retrieved passages or related facts

	HOTPOTQA		
	R@2	R@10	R@20
MDR	65.2	77.5	80.2
GraphMDR (present study)	**71.3**	**80.4**	**88.6**

Table 7 provides a comparison of the MDR and PullNet models and the proposed GraphMDR model. The MDR model, which is an MRC model, is evaluated based on a set of supporting passages, while the two hybrid models PullNet and GraphMDR are evaluated based on a supporting set, including passages and triples.

As mentioned in the previous sections, the proposed GraphMDR model provides better results in extracting the final answer. Table 7 shows that the proposed model has significant results in inferring the answer (see Sup column) as well as the correct extraction of the answer and its inference (see Joint column) compared to the other two models.

6.3. Error Analysis

Although GraphMDR outperforms other systems, it is not error-free. Examining the errors usually can improve the system performance and show the path for future researches. For GraphMDR, three general categories of errors were identified as:

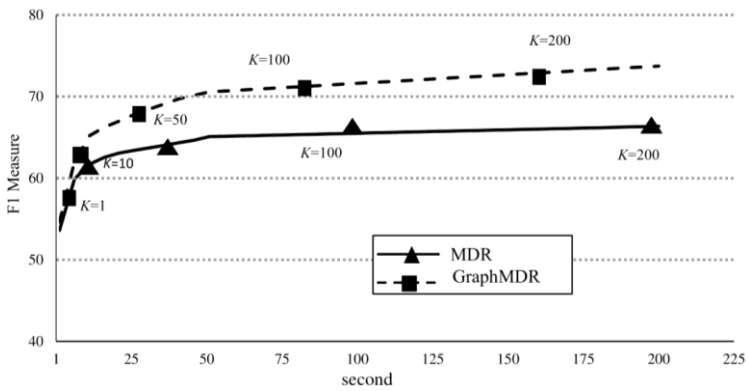

Figure 6: Comparison of MDR and GraphMDR based on the efficiency-performance trade-off.

Table 7 : Comparison of PullNet, GraphMDR and MDR based on Exact Match and Joint metrics using HotpotQA (dev)

Model	Ans				Sup				Joint			
	EM	F1	Prec	Recall	EM	F1	Prec	Recall	EM	F1	Prec	Recall
PullNet	32.1	48.5	52.5	49.6	21.2	51.2	54.2	51.8	19.1	38.2	42.1	39.1
GraphMDR	**48.2**	**61.2**	**63.2**	**64.5**	**34.3**	**65.7**	**69.4**	**63.2**	**38.4**	**51.3**	**54.8**	**51.6**
MDR	45.2	55.4	58.1	55.9	31.2	62.3	68.5	63.1	28.2	42.2	43.9	41.7

– Syntactic errors: The first category of errors that are visible in the process of extracting the answer is the result of parsing the question. Since the decomposition method used in this research is based on dependency relations, any errors in the parser leads to the wrong result in generating the sub-questions. For example, in question $\mathbf{q_2}$ in Table 10 of the appendix A.2 section, the question is divided into two sub-questions. The dependence of the word "headquartered" is incorrectly recognized to the second sub-question $\mathbf{sq_2}$, while it should have been to the first sub-question $\mathbf{sq_1}$.

– Dataset preparation errors: The second category of errors is due to the method used to generate the supporting triples, which reduces the efficiency of the proposed system. To compare the proposed method with the base system, the HOTPOTQA data set is used that has supporting passages (called **sp**). In the current research to evaluate the proposed method, for each supporting passage, supporting triples (called **sf**) is also generated. The production of triples is done by comparing the embedding vector of the supporting passage with the triple vectors of the knowledge base and finding similar vectors.

However, in several cases, all or some the supporting triples produced for a passage, are not related to the answer. Since for the evaluation, the comparison between the supporting triples and the responses extracted by the proposed method is performed, the inaccuracy of the supporting triples in the data set shows a decrease in the accuracy of the proposed method in retrieving information pieces containing responses.

For example, for the $\mathbf{q_3}$ question in Table 10 of appendix A.2, for the sub-question $\mathbf{sq_1}$, the supporting passage $\mathbf{sp_1}$ is generated in the HOTPOTQA dataset. We have generated the supporting triples $\mathbf{sf_1}$, $\mathbf{sf_2}$, and $\mathbf{sf_3}$. The correct answer to the sub-question $\mathbf{sq_1}$ is the first supporting triple, $\mathbf{sf_1}$. The second and third triples are incorrect. These two incorrect triples are generated only because of the similarity to the supporting passage $\mathbf{sp_1}$.

Although, GraphMDR works correctly and extracts the first supporting triple in the response, not extracting the other two, shows a decrease in the accuracy. To correct these errors, the method of generating the supporting triplets should be modified. In future work, the method of generating supporting passages in the HOTPOTQA dataset should be used to generate supporting triples. In the HotpotQA dataset, some people are employed to distinguish the supporting passages from the passages pool to achieve the correct answer.

– Semantic errors: The third category of errors rises from the complexity of the question concept. Answering the complex questions requires additional background knowledge. For example, in question $\mathbf{q_1}$ in Table 10 of the appendix A.2 section, "What distinction is held by …?", the answer needs to find a distinction for the entity in the question among other people. The concept of "shortest person" must be extracted as the answer. In the supporting passage, a number "5 ft" is mentioned as the height and the title, "The shortest player ever to play in the National Basketball Association" for the entity. Recognizing "The shortest height" as a distinguishing feature of an entity requires additional background knowledge.

7. Discussion

This section highlights the findings, and theoretical and practical implications of current research.

7.1. Findings

As stated in the proposed method, to extract the answers to multi-hop questions, an iterative method, searching for the number of information snippets, including intermediate answers in each step (hop), is required. The findings of this study are as follows:

– Examining the HOTPOTQA and MetaQA data sets shows that the number of steps to extract the answer is equal to the number of sub-questions (disjunctive and conjunctive sub-questions are also searched in one step). GraphMDR uses this feature for extracting the answer, so the accuracy of the proposed method is better than the base methods (Note the results reported in Table 5 and Fig. 6.). This is in contrast with the PullNet system, where there are no limits for the number of steps to find the answer.

– In existing hybrid systems, using the neural network method to extend graph nodes acts like a black box. In these methods, it is not possible to infer how to arrive at the answer (explanation for the answer). The second finding is that using a probabilistic method, enables us to extract explanations for the answer in a hybrid system. In the real world, extracting the explanations of how to reach to the answer is essential, especially in multi-hop questions. The extracted explanations contain sequences of information (including texts and triples) containing intermediate answers.

– The last finding in this study reflects the increase in system performance in the multi-shot view in comparison to the single-shot view, considering multi-hop questions. In this study, considering the question as a set of sub-questions (looking at the question in a few shots) led to an increase in the system performance, while the keeping accuracy to be competitive. In the state-of-the-art systems, not using sub-questions (one-shot look at the question) has created to an increase in search space and thus reduced the speed of extracting the answer.

7.2. Theoretical implications

The current research falls into the large field of information extraction and sub-branch of QA systems. Studies in this field have reached maturity in two main categories: finding answers from text and finding answers from knowledge base. Hybrid QA systems take advantage of both KB and text sources to extract answers. However, few hybrid systems could simultaneously use these two sources to extract answers (early fusion model).

Also, recent researches in hybrid QA, has tended to process multi-hop questions. To address the problem of finding answers to such questions, graph expansion methods have been used frequently. However, these systems extract passages from a small pool of documents to retrieve textual information using traditional methods. Using these methods, limits the speed of information extraction. In addition, the answer can only be extracted in the form of an entity of KB triples, while in many questions, the requested information is not available in the knowledge base and must be extracted from the text.

To solve passage retrieval, the theoretical concepts in the field of information retrieval prompted us to use the dense retrieval method. These methods have recently proven their success over traditional methods (TF-IDF).

Another challenge in the state-of-the-art hybrid QA systems is the limitation in extracting explanation for the answer and finding the best sequence of answers. Considering for the problem of answering multi-hop questions in hybrid QA approaches, with the idea of the sequential nature of such questions, suggested the use of a probabilistic method in solving the challenge with a well-studied text-based QA system.

This method is based on finding the best sequence of intermediate answers in the text body to reach the final question, which in the present study has been customized to create the ability to extract the best sequence of nodes of passages and entities in the graph for selecting the best answers.

In the base systems, the multi-hop question is considered as an information unit for extracting information. However, due to the sequential nature of multi-hop questions, it seems logical that the information in the question should be used in a multi-step process in the answer extraction process.

In the proposed method, the idea of using sub-questions is induced from what happens in a human mind when faced with a multi-hop question. Therefore, in the proposed method, the question is first decomposed into sub-questions. The final answer is then searched by a graph of passage, entity, and triple nodes. A probabilistic method is used to find the best sequence of answers or explanation for the final answer.

7.3. Practical implications

In general, in an information extraction system that aims to meet the information needs of users, several main important factors are involved: response time, the accuracy of response, and the explanation of how the response is extracted. Therefore, QA systems try to provide the existing challenges in improving these four essentials. According to the findings of this study, the efficiency of the proposed method can be expressed in the following cases according to the main factors mentioned.

– The use of sub-questions reduces the search space, which increases the speed of extracting the answer.

– Since the number of search steps according to the findings is equal to the number of sub-questions, there is no need for further search, this leads to an increase in the speed. It should be noted that disjunctive and conjunctive sub-questions can be searched in one step.

– As mentioned, determining how to reach the answer is a real-world necessity in information

extraction systems. The proposed system could provide the required explanation to the user by extracting the best sequence of information pieces containing intermediate answers.

8. Conclusion and Future Work

Open-domain QA systems have a long history in both text-based and knowledge-based domains. Considering the pitfalls and merits of separately extracting information from these two sources, have led the QA community to lean toward using both at the same time.

The proposed systems are still in their infancy, especially for multi-hop questions. Goals are search space reduction, explainability of the answer and the ability to extract the answer either in the form of an entity or a piece of raw text. The aim of the current research is to solve the mentioned challenges related to the two base systems of PullNet, a hybrid QA, and MDR, a text-based QA. The results show that, by comparing the proposed method with these systems and providing the possibility of extracting responses from both textual sources and a KB, the response extraction speed rises by reducing the search space, while accuracy either remains competitive or is enhanced.

Future works includes changing the method of model training so that it is independent of the KB. In addition, considering the priority of extracting sentences or similar entities, weighing them, and calculating their impacts on the accuracy can also be examined in future studies. Another plan is to examine the constraint types, especially those needing calculation, and prioritize their execution over text or a KB.

References

[1] A. Talmor, J. Berant, The web as a knowledge-base for answering complex questions, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2018) 641–651. doi:10.18653/v1/n18- 1059.

[2] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, W.W. Cohen, Open domain question answering using early fusion of knowledge bases and text, Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018. (2020) 4231–4242. doi:10.18653/v1/d18-1455.

[3] H. Sun, T. Bedrax-Weiss, W.W. Cohen, PullNet: Open domain question answering with iterative retrieval on knowledge bases and text, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. (2019) 2380–2390.

[4] M. Raison, P.E. Mazaré, R. Das, A. Bordes, Weaver: Deep Co-encoding of questions and documents for machine reading, CoRR abs/1804.10490 (2018).

[5] A. Abujabal, M. Riedewald, M. Yahya, G. Weikum, Automated template generation for question answering over knowledge graphs, 26th Int. World Wide Web Conf. WWW 2017 (2017) 1191–1200. doi:10.1145/3038912.3052583.

[6] S. Shin, K.H. Lee, Processing knowledge graph-based complex questions through question decomposition and recomposition, Information Sciences (2020) 234–244. doi:10.1016/j.ins.2020.02.065.

[7] K. Xu, L. Wu,Z. Wang, M. Yu, L. Chen, V. Sheinin, Exploiting Rich Syntactic Information for Semantic Parsing with Graph-to-Sequence Model. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018) 918-924. doi: 10.18653/v1/D18-1110.

[8] N. Bhutani, X. Zheng, H. V. Jagadish, Learning to answer complex questions over knowledge bases with query composition, Int. Conf. Inf. Knowl. Manag. Proc. (2019) 739–748. doi:10.1145/3357384.3358033.

[9] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, J. Zhao, An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge, ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (2017) 221–231. doi:10.18653/v1/P17-1021.

[10] S. HHMin, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, Multi-hop reading comprehension through question decomposition and rescoring, ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 6097–6109, 2020, doi: 10.18653/v1/p19-1613.

[11] T. Noraset, L. Lowphansirikul, and S. Tuarob, WabiQA: A Wikipedia-Based Thai Question-Answering System, Inf. Process. Manag., vol. 58, no. 1, p. 102431, 2021, doi: 10.1016/j.ipm.2020.102431.

[12] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, J. Lin, End-to-End Open-Domain Question Answering with BERTserini. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (2019) 72-77.

[13] Y. Luan, J. Eisenstein, K. Toutanova, M. Collins, Sparse, Dense, and Attentional Representations for Text Retrieval, transactions of the Association for Computational Linguistics 9 (2021) 329-345. doi:10.1162/tacl_a_00369.

[14] V. Karpukhin, O. Barlas, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W. Yih, Dense Passage Retrieval for Open-Domain Question Answering, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020) 6769-6781. doi: 10.18653/v1/2020.emnlp-main.550.

[15] W. Xiong et al., Answering complex open-domain questions with multi-hop dense retrieval, International Conference on Learning Representations, 2020.

[16] L. Song, Z. Wang, M. Yu, Y. Zhang, R. Florian, D. Gildea, Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks, arXiv:1809.02040 (2018).

[17] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electron., vol. 8, no. 8, pp. 1–34, 2019, doi: 10.3390/electronics8080832.

[18] T. Miller, Explanation in Artificial Intelligence : Insights from the Social Sciences, Artificial intelligence, 2019, 267: 1-38 .

[19] R. Das et al., Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning, International Conference on Learning Representations, 2017.

[20] W. Zheng, J.X. Yu, L. Zou, H. Cheng, Question Answering Over Knowledge Graphs: Question Understanding Via Template Decomposition, Proceedings of the VLDB Endowment 11.11 (2018) 1373-1386. doi:10.14778/3236187.3236192.

[21] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, Variational Reasoning for Question Answering with Knowledge Graph. Thirty-Second AAAI Conf. Artif. Intell., 2018.

[22] M. Yu, W. Yin, K. S. Hasan, C. dos Santos, B. Xiang, B. Zhou, Improved Neural Relation Detection for Knowledge Base Question Answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2017) 571-581.

[23] K. Luo, F. Lin, X. Luo, K. Zhu, Knowledge base question answering via encoding of complex query graphs. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018) 2185–2194. doi:10.18653/v1/d18-1242.

[24] L. Dong and M. Lapata, Language to logical form with neural attention, 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap., vol. 1, pp. 33–43, 2016, doi: 10.18653/v1/p16-1004.

[25] C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao, Neural symbolic machines: Learning semantic parsers on freebase with weak supervision, ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 1, pp. 23–33, 2017, doi: 10.18653/v1/P17-1003.

[26] X. Yao and B. Van Durme, Information Extraction over Structured Data: Question Answering with Freebase Center for Language and Speech Processing," Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Volume 1 Long Pap., pp. 956–966, 2014.

[27] B. Fu, Y. Qiu, C. Tang, Y. Li, H. Yu, and J. Sun, "A survey on complex question answering over knowledge base: Recent advances and challenges," arXiv, no. 1, pp. 1–19, 2020.

[28] Y. Chen, L. Wu, and M. J. Zaki, Bidirectional Attentive Memory Networks for Question Answering over Knowledge Bases," Proceedings of NAACL-HLT. 2019. p. 2913-2923.

[29] Y. Qiu, Y. Wang, X. Jin, and K. Zhang, Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision, WSDM 2020 - Proc. 13th Int. Conf. Web Search Data Min., pp. 474–482, 2020, doi: 10.1145/3336191.3371812.

[30] N. Bhutani, X. Zheng, K. Qian, Y. Li, H. Jagadish, Answering Complex Questions by Combining Information from Curated and Extracted Knowledge Bases, Proceedings of the First Workshop on Natural Language Interfaces (2020) 1–10. doi:10.18653/v1/2020.nli-1.1.

[31] D. Gillick et al., Learning dense representations for entity retrieval, CoNLL 2019 - 23rd Conf. Comput. Nat. Lang. Learn. Proc. Conf., pp. 528–537, 2019, doi: 10.18653/v1/k19-1049.

[32] G. Izacard, E. Grave, Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: . (2020) 874-880.

[33] Z. A. Taeb and S. Momtazi, Text-based question answering from information retrieval and deep neural network perspectives: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2021, 11.6: e1412

[34] H. K. Azad and A. Deepak, Query expansion techniques for information retrieval: A survey, Inf. Process. Manag., vol. 56, no. 5, pp. 1698–1735, 2019, doi: 10.1016/j.ipm.2019.05.009.

[35] X. Lu, R.S. Roy, Y. Wang, G. Weikum, A. Alexa, Answering Complex Questions by Joining Multi-Document Evidence with Quasi Knowledge Graphs, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019) 105-114. doi:10.1145/3331184.3331252.

[36] D. Savenkov and E. Agichtein, When a knowledge base is not enough: Question answering over knowledge bases with external text data, SIGIR 2016 - Proc. 39th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 235–244, 2016, doi: 10.1145/2911451.2911536.

[37] K. Xu, S. Reddy, Y. Feng, S. Huang, and D. Zhao, Question answering on freebase via relation extraction and textual evidence, 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap., vol. 4, pp. 2326–2336, 2016, doi: 10.18653/v1/p16-1220.

[38] *B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull, S. Gupta, Y. Mehdad, S. Yih, Unified Open-Domain Question Answering with Structured and Unstructured Knowledge, arXiv:2012.14610 (2020).*

[39] R. Das, M. Zaheer, S. Reddy, and A. McCallum, "Question answering on knowledge bases and text using universal schema and memory networks," ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 2, pp. 358–365, 2017, doi: 10.18653/v1/P17-2057.

[40] M. Thayaparan, M. Valentino, and A. Freitas, "A Survey on Explainability in Machine Reading Comprehension, arXiv preprint arXiv:2010.00389, 2020.

[41] Z. Yang et al., "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 2369–2380, 2020, doi: 10.18653/v1/d18-1259.

[42] P. Jansen, N. Balasubramanian, M. Surdeanu, and P. Clark, "What ' s in an Explanation ? Characterizing Knowledge and Inference Requirements for Elementary Science Exams." In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 2956-2965), 2016.

[43] H. Ji, J. Nothman, and B. Hachey, Overview of TAC-KBP2014 Entity Discovery and Linking Tasks, In: Proc. Text Analysis Conference (TAC2014). 2014. p. 1333-1339.

[44] K. Balog, Entity Linking, Entity-Oriented Search. Springer, Cham, 2018. 147-188.

[45] P. Cifariello, P. Ferragina, and M. Ponza, W ISER : A Semantic Approach for Expert Finding in Academia based on Entity Linking, Information Systems, 2019, 82: 1-16.

[46] A. H. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, and J. Weston, Key-value memory networks for directly reading documents, EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 1400–1409, 2016, doi: 10.18653/v1/d16-1147.

[47] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, TriviaQA: A large scale distantly supervised challenge

dataset for reading comprehension, ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 1, pp. 1601–1611, 2017, doi: 10.18653/v1/P17-1147.

[48] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, SQuad: 100,000+ questions for machine comprehension of text, EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., no. ii, pp. 2383–2392, 2016, doi: 10.18653/v1/d16-1264.

[49] W. T. Yih, M. Richardson, C. Meek, M. W. Chang, and J. Suh, The value of semantic parse labeling for knowledge base question answering, 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Short Pap., pp. 201–206, 2016, doi: 10.18653/v1/p16-2033.

[50] S. Asadifar, M. Kahani, and S. Shekarpour, Schema and content aware classification for predicting the sources containing an answer over corpus and knowledge graphs, PeerJ Computer Science, 2022, 8: e846, pp. 1–29, doi: 10.7717/peerj-cs.846.

[51] X. Yin, Y. Huang, B. Zhou, A. Li, L. Lan, and Y. Jia, Deep Entity Linking via Eliminating Semantic Ambiguity With BERT, IEEE Access, vol. 7, pp. 169434–169445, 2019, doi: 10.1109/ACCESS.2019.2955498.

[52] M. Ding, C. Zhou, Q. Chen, H. Yang, J. Tang, and A. Group, Cognitive Graph for Multi-Hop Reading Comprehension at Scale, arXiv preprint arXiv:1905.05460, 2019 2017.

Appendix A

A.1: Running Examples throughout the program

This section provides examples of multi-hop questions from two datasets HOTPOTQA and MetaQA that are referenced throughout the article to understand the proposed method.

A.1.1: HOTPOTQA dataset

Table 9 contains examples of the HOTPOTQA data set questions that are referenced in the article for clarification.

A.1.2 MetaQA dataset

In this section, two examples of 2-step and 3-step questions from the data set are given. The questions were created using two and three patterns that are specified in the table as type. The sub-questions for each question are generated by the proposed approach in this paper and, as can be seen, correspond to the patterns used to construct the questions.

A.2. Error Cases in present approach, GraphMRD

The examples in Table 10 are selected from the HOTPOTQA dataset to interpret the challenges faced by the proposed method. The q1, q2, and q3 questions are presented as examples of errors in concept complexity, question decomposition, and the method of generating supporting triples, respectively

Table 8: Examples of 2 and 3-hop questions from the MetaQA dataset along with the fsub-questions generated by the proposed method.

2-hop	**q1**: who are the directors of the films written by [Laura Kerr]? **answer**: H.C. Potter. **qtype**: writer_to_movie_to_director. **sq1**: which films written by [Laura Kerr]? **sq2**: who are the directors of the films?
3-hop	**q2**: what types are the films directed by the director of [For Love or Money]? **answer**: Action|Comedy|Western|Thriller|Crime. **qtype**: Movie-to-director-to-movie-to-genre. **sq1**: director of [For Love or Money]? **sq2**: the films directed by the director? **sq3**: what types are the films?

.

Table 9: Examples of HOTPOTQA datasets. sq_i,sp_j, and sf_k stand for i-th sub-question, j-th supporting passage and k-th supporting fact for question qt, respectively.

q_1: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?
answer: Chief of Protocol.

sq_1: Which woman who portrayed Corliss Archer in the film Kiss and Tell?
sp_1: (Kiss and Tell (1945 film)) Kiss and Tell is a 1945 American comedy film starring then 17-year-old Shirley Temple as Corliss Archer.
sf_1: not existed.

sq_2: What government position was held by the woman?
sp_1: (Shirley Temple) Shirley Temple Black (April 23, 1928 – February 10, 2014) was an American actress, singer, dancer, businesswoman, and diplomat who was Hollywood's number one box-office draw as a child actress from 1934 to 1938.
sf_1: (dbr: Shirley Temple, dbo:wikiPageWikiLink, dbc:American film actresses).
sp_2: (Shirley Temple) Shirley Temple As an adult, she was named United States ambassador to Ghana and to Czechoslovakia, and also served as Chief of Protocol of the United States.
sf_2: (dbr: Shirley Temple, dct:subject, dbc:Chiefs of Protocol of the United States).

q_2: In what city did the \"Prince of tenors\" star in a film based on an opera by Giacomo Puccini?
answer: Rome.

sq_1: Which film is based on an opera by Giacomo Puccini?
sp_1: (Tosca (1956 film)) It is based on the 1900 opera Tosca by Giacomo Puccini, which was adapted from the 1887 play by Victorien Sardou.
sf_1: not existed.

sq_2: In what city did the \"Prince of tenors\" star in a film?
sp_1: (Tosca (1956 film)) It was made at Cinecittà in Rome.
sf_1: not existed

q_3: Ralph Hefferline was a psychology professor at a university that is located in what city?
answer: New York City

sq_1: Ralph Hefferline was a psychology professor at what university?
sp_1: (Ralph Hefferline) Ralph Franklin Hefferline (15 February 1910 in Muncie, Indiana – 16 March 1974) was a psychology professor at Columbia University.
sf_1: not existed.

sq_2: the university is located in what city?
sp_1: (Columbia University) Columbia University (also known as Columbia, and officially as Columbia University in the City of New York) is a private Ivy League research university in New York City.
sf_1: (dbr: Columbia University, dbp:city, dbr:New York City).

Table 10: Examples of HotpotQA datasets for Error Cases in present approach. sq_i,sp_j, and sf_k stand for i-*th* sub-question, j-*th* supporting passage and k-*th* supporting fact for question q_i, respectively.

q1: What distinction is held by the former NBA player who was a member of the Charlotte Hornets during their 1992-93 season and was head coach for the WNBA team Charlotte Sting?
answer: shortest player ever to play in the National Basketball Association

sq1: Which NBA player who was a member of the Charlotte Hornets during their 1992-93 season and was head coach for the WNBA team Charlotte Sting?

sp1: (1992–93_Charlotte_Hornets_season) With the addition of Mourning, along with second-year star Larry Johnson and Muggsy Bogues, the Hornets struggled around .500 for most of the season, but won 9 of their final 12 games finishing their season third in the Central Division with a 44–38 record, and qualified for their first ever playoff appearance.

sf1: (dbr: 1992 93 Charlotte Hornets season, dbo:wikiPageWikiLink, dbr:Muggsy Bogues).

sp2: (Muggsy Bogues) After his NBA career, he served as head coach of the now-defunct WNBA team Charlotte Sting.

sf2: (dbr:Muggsy Bogues, is dbp:coach of, dbr:2006 Charlotte Sting season).

sq2: What distinction is held by the former NBA player?

sp1: (Muggsy Bogues) The shortest player ever to play in the National Basketball Association, the 5 ft Bogues played point guard for four teams during his 14-season career in the NBA.

sf1: not existed.

q2: Where is the company that Sachin Warrier worked for as a software engineer headquartered?
answer: Ronald Shusett.

sq1: Which company that Sachin Warrier worked for as a software engineer headquartered?

sp1: (Sachin Warrier) He was working as a software engineer in Tata Consultancy Services in Kochi.

sf1: (dbr: Tata Consultancy Services, is dbo:wikiPageWikiLink of, dbr:Kochi).

sq2: Where is the company?

sp1: (Tata Consultancy Services) Tata Consultancy Services (TCS) is an Indian multinational information technology (IT) services and consulting company, headquartered in Mumbai, Maharashtra, India and largest campus and workforce in Chennai, Tamil Nadu, India.

sf1: (dbr:Tata Consultancy Service, dpp:located, dbr:Mumbai).

q3: What is the name of the fight song of the university whose main campus is in Lawrence, Kansas and whose branch campuses are in the Kansas City metropolitan area?
answer: Kansas Song.

sq1: Which university whose main campus is in Lawrence, Kansas?

sp1: (University of Kansas) The main campus in Lawrence, one of the largest college towns in Kansas, is on Mount Oread, the highest elevation in Lawrence.

sf1: (dbr: University of Kansas, dbo:city, dbr:Lawrence Kansas).

sf2: (dbr: Mount Oread, dbo:locatedInArea, dbr:Lawrence Kansas).

sf3: (Mount Oread, dbp:elevationFt, 1037).

sq2: Which university whose branch campuses are in the Kansas City metropolitan area?

sp1: (University of Kansas) Two branch campuses are in the Kansas City metropolitan area: the Edwards Campus in Overland Park, and the university's medical school and hospital in Kansas City.

sf1: (dbr: University of Kansas, dbo:city, dbr:Lawrence Kansas).

sf2: (dbr: University of Kansas, dbo:wikiPageWikiLink, dbr:University of Kansas Medical Center).

sq3: What is the name of the fight song of the university?

sp1: (Kansas Song) :Kansas Song (We're From Kansas) is a fight song of the University of Kansas.

sf1: (dbr:Kansas Song, dbo:wikiPageWikiLink, dbr:University of Kansas).